TCGA-Assembler Quick Start Guide

Yitan Zhu[1], Yuan Ji[1,2]

1. Center for Biomedical Research Informatics, NorthShore University HealthSystem, Evanston, IL 60201
2. Department of Health Studies, The University of Chicago, Chicago, IL 60637

Email: zhuyitan@gmail.com

Feb. 10th, 2014

1. Introduction

This guide provides a quick start to use TCGA-Assembler. It concisely demonstrates most of the data retrieval and processing functions in TCGA-Assembler, although the full set of functions and their details are provided in TCGA-Assembler User Manual. R commands in this guide can be directly copied and pasted for testing in R environment on user's computer.

CAUTION: We identified a defect in Mac application "Preview.app". Do not open this pdf file in Preview and copy/paste the R code for testing. Instead use Adobe reader (http://get.adobe.com/reader/) or Acrobat to open this pdf file and copy/paste code to R for testing.

2. System Requirements

Downloading and processing TCGA data from internet may need significant memory space depending on the size of data to be retrieved and processed, we recommend using TCGA-Assembler on computers with 16GB or larger RAM and with a fast and stable Internet connection. However, all the examples in this guide should work well on computers with 4GB~8GB RAM, as we only use small datasets to demo the functions. Also users need to have basic knowledge of R to use TCGA-Assembler.

TCGA-Assembler is built on R (http://www.r-project.org/) and requires R packages HGNChelper, RCurl, httr, stringr, digest, bitops, and their dependents. We assume that users have a recent R version installed (version 2.15.1 or later). To start, users should launch R and install the required packages, for example using command

```
install.packages(c("HGNChelper", "RCurl", "httr", "stringr", "digest", "bitops"), dependencies=T)
```

***Remark**: R can be downloaded and installed from http://www.r-project.org/. Another way to install the R packages is that in R GUI (Graphical User Interface), go to Packages menu and click on Install package(s). Select the best CRAN mirror site for you. And then select the package and click ok to install.*

***Caution1**: Depending on which R packages are already installed on users' computers, occasionally users could experience R errors complaining about conflicts of different packages. We did not experience any in our testing, but do not rule out potential errors due to system setup, or clashing with existing R packages.*

***Caution2**: When using TCGA-Assembler, avoid reading or writing files in Dropbox or similar cloud-based folders, which may produce error message. Make sure that you have read and write access to the working directory.*

3. A Two-Minute Example

3.1 Installation and Configuration

To download and use TCGA-Assembler, go to http://health.bsd.uchicago.edu/yji/TCGA-Assembler.htm. Click Download Software and unzip the downloaded file to your desired file directory on the local computer. For example, in our own test, we unzipped the package and created the folder /Users/zhuy/TCGA-Assembler/ for the unzipped files. Then we set the Present Working Directory (PWD) of R using

```
setwd("/Users/zhuy/TCGA-Assembler")
```

Users should use

```
setwd("foo")
```

where foo is the directory that stores unzipped TCGA-Assembler files on user's computer.

3.2 A Quick Example

TCGA-Assembler includes two modules, Module A and Module B. Functions in Module A retrieve data from TCGA Data Coordinating Center (DCC), and functions in Module B process and integrate the retrieved data. The main utilities of TCGA-Assembler include 1) data retrieval, 2) data process, and 3) data integration. To demonstrate all three utilities we provide an example in which we download, process, and combine RNA-Seq and DNA copy number data for several patient samples of rectum adenocarcinoma (READ). The example below takes about 2 minutes on author's computer with 2.93GHz CPU and Internet speed of 2.4 MB per second.

Step 1. Load all the functions in Modules A and B into the working space.

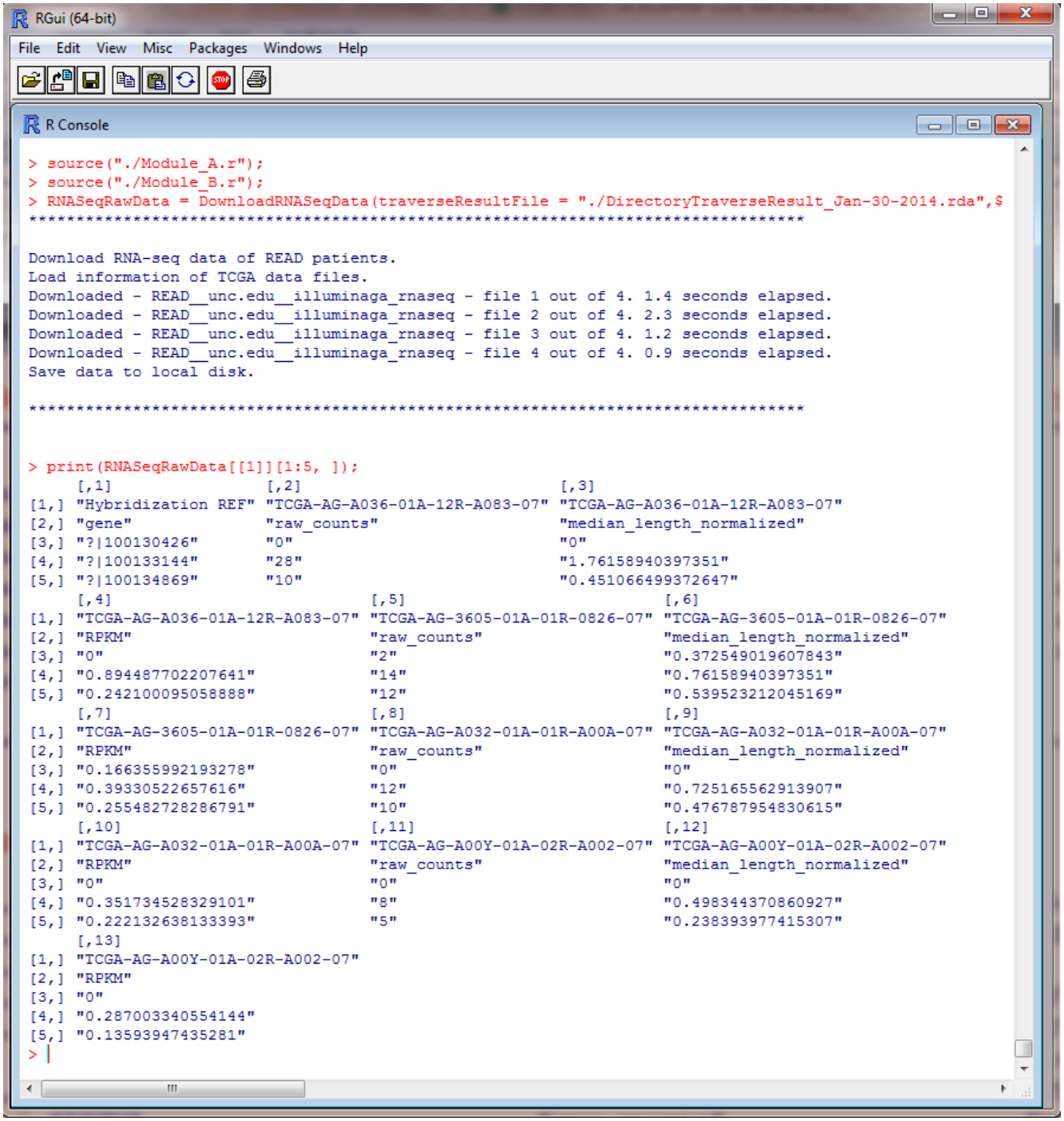
```
source("./Module_A.r");
source("./Module_B.r");
```

Step 2. Retrieve RNA-Seq gene expression data and DNA copy number data of several READ samples (with specified TCGA barcodes) from TCGA DCC website. Use the following command to download RNA-seq gene expression data of four patient samples and then look at the top 5 rows of the data.

```
RNASeqRawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform = "RNASeqV1", dataType = "gene.quantification", inputPatientIDs = c("TCGA-AG-A036-01", "TCGA-AG-3605-01", "TCGA-AG-A032-01", "TCGA-AG-A00Y-01"), outputFileName = "2minuteExample");

print(RNASeqRawData[[1]][1:5, ]);
```

The following screenshot is obtained after executing the commands.

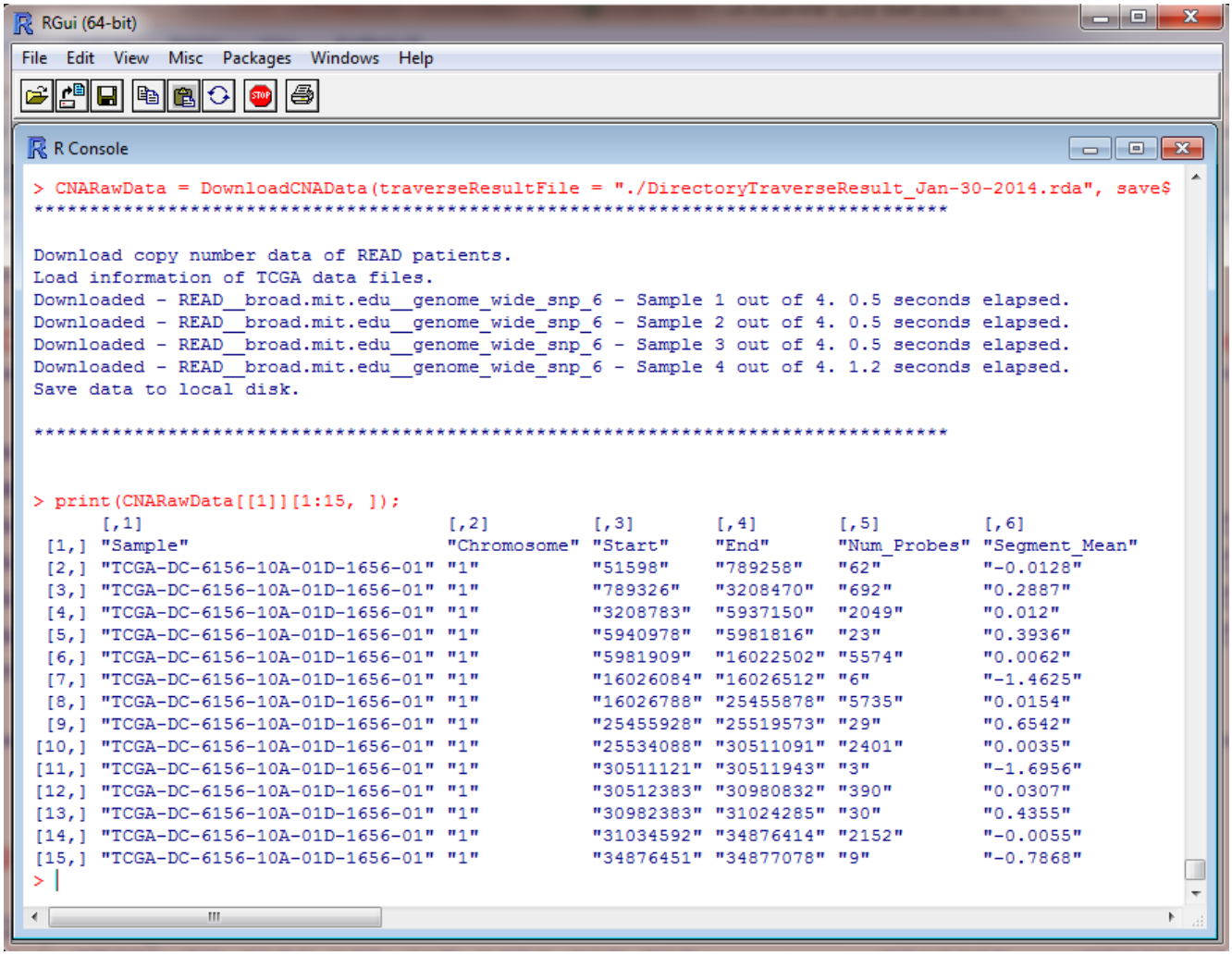

RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> source("./Module_A.r");
> source("./Module_B.r");
> RNASeqRawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda",$
****
Download RNA-seq data of READ patients.
Load information of TCGA data files.
Downloaded - READ__unc.edu__illuminaga_rnaseq - file 1 out of 4. 1.4 seconds elapsed.
Downloaded - READ__unc.edu__illuminaga_rnaseq - file 2 out of 4. 2.3 seconds elapsed.
Downloaded - READ__unc.edu__illuminaga_rnaseq - file 3 out of 4. 1.2 seconds elapsed.
Downloaded - READ__unc.edu__illuminaga_rnaseq - file 4 out of 4. 0.9 seconds elapsed.
Save data to local disk.
****
> print(RNASeqRawData[[1]][1:5,]);
[,1] [,2] [,3]
[1,] "Hybridization REF" "TCGA-AG-A036-01A-12R-A083-07" "TCGA-AG-A036-01A-12R-A083-07"
[2,] "gene" "raw_counts" "median_length_normalized"
[3,] "?|100130426" "0" "0"
[4,] "?|100133144" "28" "1.76158940397351"
[5,] "?|100134869" "10" "0.451066499372647"
[,4] [,5] [,6]
[1,] "TCGA-AG-A036-01A-12R-A083-07" "TCGA-AG-3605-01A-01R-0826-07" "TCGA-AG-3605-01A-01R-0826-07"
[2,] "RPKM" "raw_counts" "median_length_normalized"
[3,] "0" "2" "0.372549019607843"
[4,] "0.894487702207641" "14" "0.76158940397351"
[5,] "0.242100095058888" "12" "0.539523212045169"
[,7] [,8] [,9]
[1,] "TCGA-AG-3605-01A-01R-0826-07" "TCGA-AG-A032-01A-01R-A00A-07" "TCGA-AG-A032-01A-01R-A00A-07"
[2,] "RPKM" "raw_counts" "median_length_normalized"
[3,] "0.166355992193278" "0" "0"
[4,] "0.39330522657616" "12" "0.725165562913907"
[5,] "0.255482728286791" "10" "0.476787954830615"
[,10] [,11] [,12]
[1,] "TCGA-AG-A032-01A-01R-A00A-07" "TCGA-AG-A00Y-01A-02R-A002-07" "TCGA-AG-A00Y-01A-02R-A002-07"
[2,] "RPKM" "raw_counts" "median_length_normalized"
[3,] "0" "0" "0"
[4,] "0.351734528329101" "8" "0.498344370860927"
[5,] "0.222132638133393" "5" "0.238939977415307"
[,13]
[1,] "TCGA-AG-A00Y-01A-02R-A002-07"
[2,] "RPKM"
[3,] "0"
[4,] "0.287003340554144"
[5,] "0.13593947435281"
>

Use the following command to download copy number data of four patient samples and then look at the top 15 rows of the data.

CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform = "genome_wide_snp_6", inputPatientIDs = c("TCGA-DC-6156-10", "TCGA-AG-3605-01", "TCGA-AG-A032-01", "TCGA-AG-A00Y-01"), outputFileName = "2minuteExample");

print(CNARawData[[1]][1:15,]);

The following figure is the screenshot obtained after executing the commands.

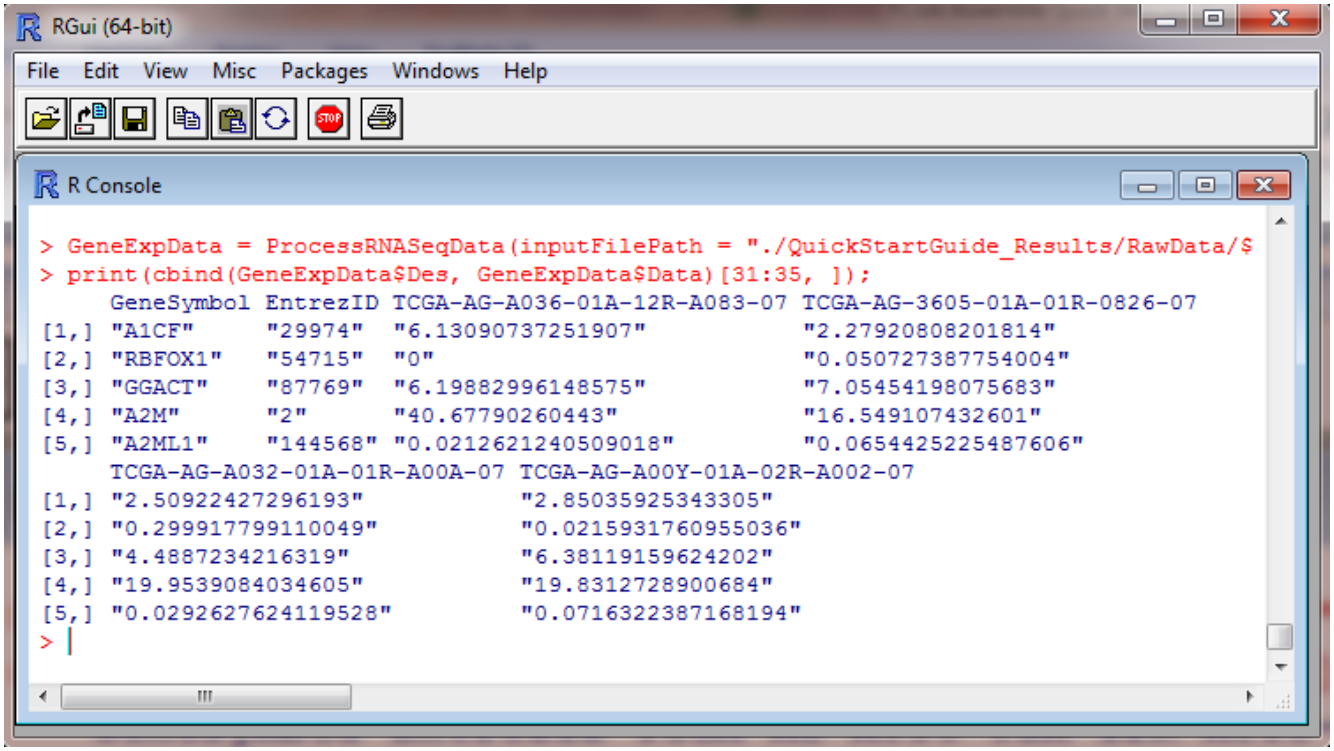

RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", save$
Download copy number data of READ patients.
Load information of TCGA data files.
Downloaded - READ__broad.mit.edu__genome_wide_snp_6 - Sample 1 out of 4. 0.5 seconds elapsed.
Downloaded - READ__broad.mit.edu__genome_wide_snp_6 - Sample 2 out of 4. 0.5 seconds elapsed.
Downloaded - READ__broad.mit.edu__genome_wide_snp_6 - Sample 3 out of 4. 0.5 seconds elapsed.
Downloaded - READ__broad.mit.edu__genome_wide_snp_6 - Sample 4 out of 4. 1.2 seconds elapsed.
Save data to local disk.
> print(CNARawData[[1]][1:15,]);
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] "Sample" "Chromosome" "Start" "End" "Num_Probes" "Segment_Mean"
[2,] "TCGA-DC-6156-10A-01D-1656-01" "1" "51598" "789258" "62" "-0.0128"
[3,] "TCGA-DC-6156-10A-01D-1656-01" "1" "789326" "3208470" "692" "0.2887"
[4,] "TCGA-DC-6156-10A-01D-1656-01" "1" "3208783" "5937150" "2049" "0.012"
[5,] "TCGA-DC-6156-10A-01D-1656-01" "1" "5940978" "5981816" "23" "0.3936"
[6,] "TCGA-DC-6156-10A-01D-1656-01" "1" "5981909" "16022502" "5574" "0.0062"
[7,] "TCGA-DC-6156-10A-01D-1656-01" "1" "16026084" "16026512" "6" "-1.4625"
[8,] "TCGA-DC-6156-10A-01D-1656-01" "1" "16026788" "25455878" "5735" "0.0154"
[9,] "TCGA-DC-6156-10A-01D-1656-01" "1" "25455928" "25519573" "29" "0.6542"
[10,] "TCGA-DC-6156-10A-01D-1656-01" "1" "25534088" "30511091" "2401" "0.0035"
[11,] "TCGA-DC-6156-10A-01D-1656-01" "1" "30511121" "30511943" "3" "-1.6956"
[12,] "TCGA-DC-6156-10A-01D-1656-01" "1" "30512383" "30980832" "390" "0.0307"
[13,] "TCGA-DC-6156-10A-01D-1656-01" "1" "30982383" "31024285" "30" "0.4355"
[14,] "TCGA-DC-6156-10A-01D-1656-01" "1" "31034592" "34876414" "2152" "-0.0055"
[15,] "TCGA-DC-6156-10A-01D-1656-01" "1" "34876451" "34877078" "9" "-0.7868"
>

Step 3. Processes the downloaded data to perform basic quality control and output clean data matrix files, where each row is a genomic feature and each column corresponds to a sample. Use the following command to process RNA-seq gene expression data and then check a few rows of the processed data. RPKM (reads per kilo base per million) values of gene expressions are extracted for subsequent analysis.

```
GeneExpData = ProcessRNASeqData(inputFilePath =
"./QuickStartGuide_Results/RawData/2minuteExample__READ__unc.edu__illuminaga_rnaseq__gene.
quantification__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_rnaseqv2__GeneExp",
outputFileFolder = "./QuickStartGuide_Results/BasicProcessingResult", dataType = "GeneExp", verType
= "RNASeqV1");

print(cbind(GeneExpData$Des, GeneExpData$Data)[31:35, ]);
```

The following screenshot shows the R console after executing the two commands.

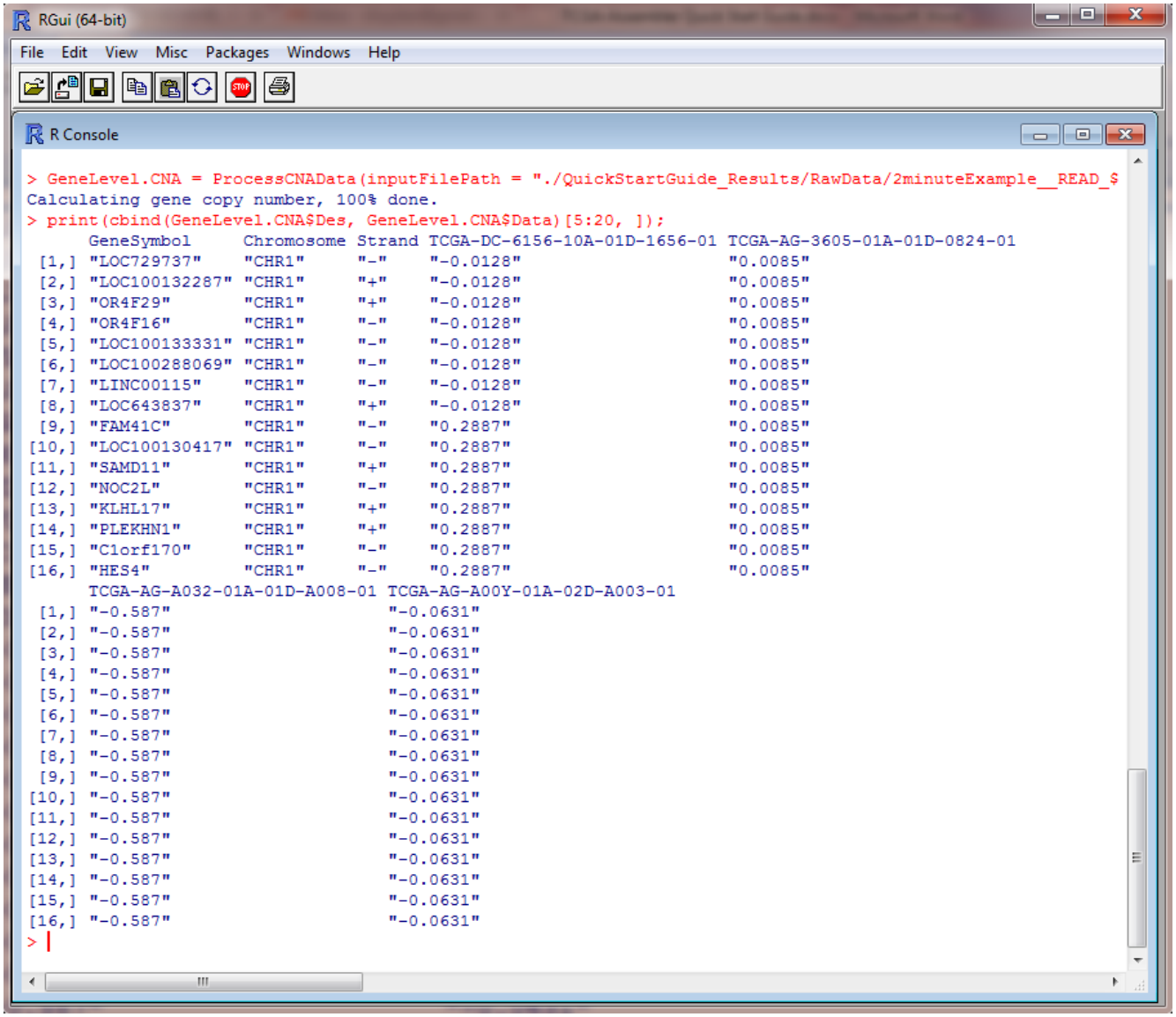


Use the following command to process downloaded copy number data and calculate copy numbers of genes. Then, check a few rows of the data.

```
GeneLevel.CNA = ProcessCNAData(inputFilePath =
"./QuickStartGuide_Results/RawData/2minuteExample__READ__broad.mit.edu__genome_wide_snp_
6__hg18__Jan-30-2014.txt", outputFileName = "READ__genome_wide_snp_6__GeneLevelCNA",
outputFileFolder = "./QuickStartGuide_Results/BasicProcessingResult", refGenomeFile =
"./SupportingFiles/Hg18GenePosition.txt");

print(cbind(GeneLevel.CNA$Des, GeneLevel.CNA$Data)[5:20, ]);
```

The following figure is the screenshot obtained after executing the commands.



Step 4. Integrate RNA-Seq and copy number data into a mega data matrix. First, form a list object containing the two processed datasets, which will be input into the data integration function.

```
dataList = vector("list", 2);
dataList[[1]] = list(Data = GeneExpData$Data, Des = GeneExpData$Des , dataType = "GeneExp");
dataList[[2]] = list(Data = GeneLevel.CNA$Data, Des = GeneLevel.CNA$Des,  dataType = "CNA");
```

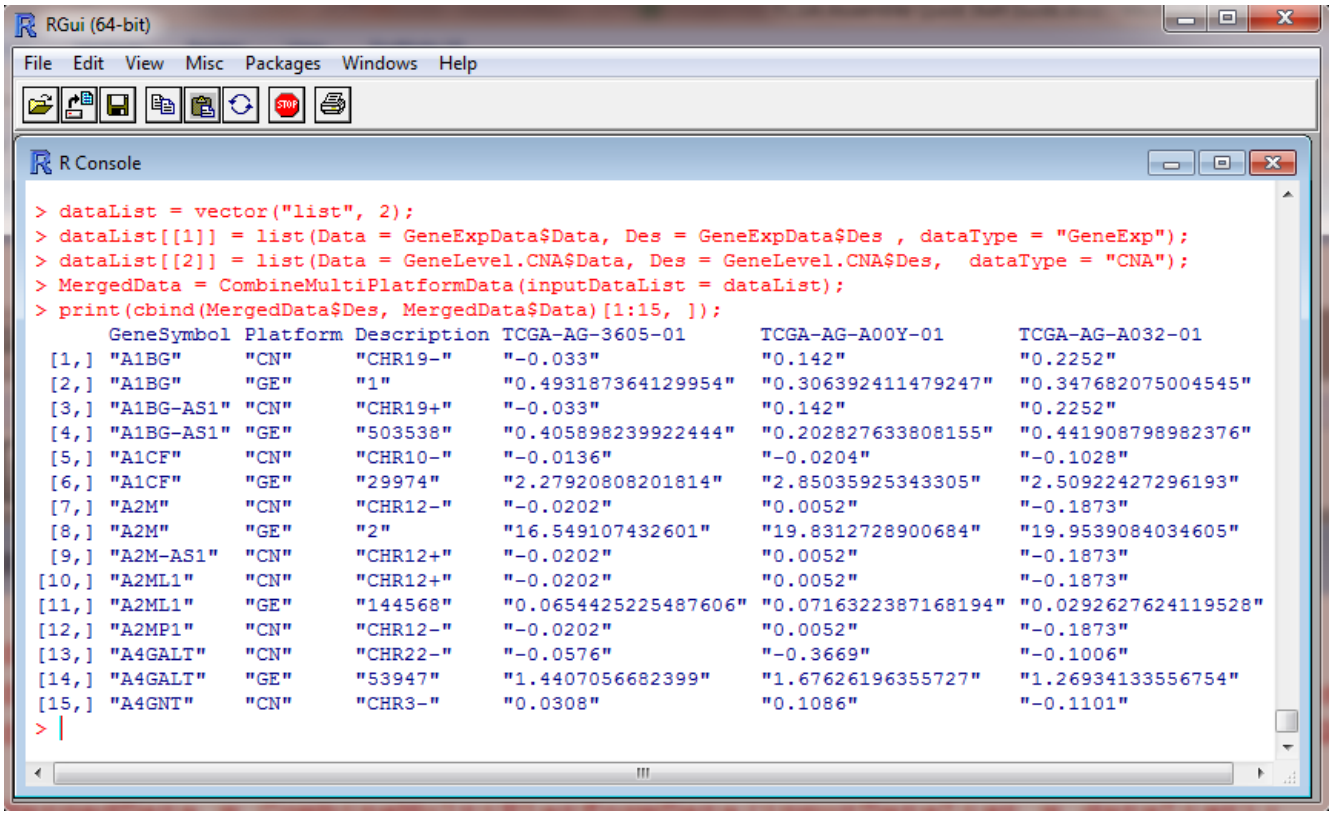
Second, use the following commands to integrate the data and then check the top 15 rows of data.

```
MergedData = CombineMultiPlatformData(inputDataList = dataList);

print(cbind(MergedData$Des, MergedData$Data)[1:15, ]);
```

The following is a screenshot obtained after executing the commands. Three samples are measured by both gene expression and copy number assays. Thus they are kept in the combined data. Gene expression (GE) value and copy number (CN) value of the same gene are adjacent rows in the combined

data. The description of CN platform is the chromosome ID and strand of gene. The description of GE platform is the Entrez ID of gene.



***Remark**: Before downloading data, TCGA-Assembler needs to gather information about all TCGA open-access data files, for every cancer type, every assay platform, and every different versions of the data. This is fulfilled by the function "TraverseAllDirectories()"that traverses all the sub-directories in the open-access HTTP directory on the data server of TCGA DCC and obtains the URLs of all data files. These URLs will be used by other functions to download various TCGA data. An example command to use the TraverseAllDirectories function is*

TraverseAllDirectories(entryPoint = "https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/", fileLabel = "DirectoryTraverseResult");

This function must be executed initially for a new user, and takes about one hour to complete as it will retrieve hundreds of thousands of URLs over Internet. In the examples of this user guide, we have completed this step and included the resulted file called "DirectoryTraverseResult_Jan-30-2014.rda" in the package. This file contains all the URLs and is used as an input argument of the data downloading functions in this user guide. In the future, users could execute the traverse function by themselves to update the URLs as TCGA continues to expand. See Section 5 for more detail about the traverse function.

4. A Twenty-Minute Example

In this section, we demonstrate an example using a pre-coded program, QuickStartGuide_Examples.r, to fully illustrate TCGA-Assembler. Details of the commands will be explained in Section 5. It will take about 20 minutes to fully execute the commands, following two simple steps.

Step 1. Start R, and change the working directory of R to the TCGA-Assembler folder that users unzipped, using the command setwd. In our case, we unzipped TCGA-Assembler to the user home directory, and therefore ran the following command. Users should change the directory name accordingly.

```
setwd("/Users/zhuy/TCGA-Assembler")
```

Step 2. Source the example R script QuickStartGuide_examples.r to execute all the commands in it.

```
source("QuickStartGuide_Examples.r")
```

Wait until the execution to complete, which takes about 20 minutes depending on your internet connection speed and CPU processing speed. During the process you will see messages in R Console showing the progress of execution and also how much time it spends.

After completion, a new folder called QuickStartGuide_Results is created in the TCGA-Assembler folder. In the QuickStartGuide_Results folder, there are three subfolders, RawData, BasicProcessingResult, and AdvancedProcessingResult. These folders contain all the result files generated from the test functions in QuickStartGuide_Examples.r. They demonstrate the data downloading, basic processing, and advance processing utilities of TCGA-Assembler. It will take some time for users to go over all the functions and their expected outcomes. We recommend that at this point, users open the R script QuickStartGuide_Examples.r in an editor and walk through the commands and our comments line by line. The rest of guide is devoted to such a walk-through.

5. Appendix: Details of Functions and Outputs

In this section, we introduce TCGA-Assembler functions that are used in the 20-minute example above. Without further explanation, we assume that users have opened the R script QuickStartGuide_Examples.r and generated results by sourcing the R script. Three result subfolders should be generated under the new folder QuickStartGuide_Results.

The RawData subfolder holds various raw data retrieved from TCGA DCC data server. Data files in the RawData subfolder are tab-delimited .txt files. The file names indicate the data contents including cancer type, institution that produced the data, assay platform used to generate the data, and data type or reference genome.

The BasicProcessingResult subfolder holds results obtained by processing the raw data using basic data processing functions in TCGA-Assembler.

The AdvancedProcessingResult subfolder holds results obtained by further processing the basic processing results using advanced data processing functions in TCGA-Assembler.

5.1 Introduction of Data Acquisition Functions in Module A.

Before downloading data, TCGA-Assembler needs to gather the information about all the TCGA open-access data files, for every cancer type, assay platform, and different versions of the data. This is fulfilled by the TraverseAllDirectories function that traverses all the sub-directories in the open-access HTTP directory on the data server of TCGA DCC and obtains the URLs of all data files. These URLs will be used by other functions to download various TCGA data. An example command to use the TraverseAllDirectories function is

```
TraverseAllDirectories(entryPoint = "https://tcga-
data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/", fileLabel =
"DirectoryTraverseResult");
```

The first input argument of the function is the URL of the root directory including all TCGA public data on DCC data server. The second input argument is a character string to form the name of the directory traverse result file that will store the URLs of all public data files and will be saved in the PWD. Due to the vast amount of sub-directories (~18,000) and files (~1,300,000), this traverse process can take about an hour to complete depending upon the Internet connection speed. *It needs to be done only once for downloading the data of all various cancer types and assay platforms.* To save users' time, we included in the package a directory traverse result file DirectoryTraverseResult_Jan-30-2014.rda, which includes the URLs of all open-access data files existing on the TCGA data server on Jan. 30th, 2014. The file URLs are usually stable and valid for quite a long time, so users can skip running the TraverseAllDirectories function and use the existing directory traverse result to download data in this quick start guide.

There are fix data acquisition functions in Module A, whose names all start with "Download", including DownloadmiRNASeqData, DownloadCNAData, DownloadRNASeqData, DownloadMethylationData, DownloadRPPAData, and DownloadClinicalData. Each of these functions downloads TCGA public data of one genomic platform for user specified cancer type. To acquire TCGA data, these functions (1) identify the URLs of the data files to be downloaded based on the directory traverse result, (2) download data files of individual samples, and (3) assemble the downloaded data files into data matrix files.

A single cancer type in TCGA usually has several hundreds of samples measured by different assay platforms. Acquiring the data of all samples can take some time. To avoid keeping users wait, in this quick start guide we use an option in the data downloading functions that allows downloading only the data of specified patient samples rather than the whole dataset including all samples. Besides the downloading time, there is no other difference between the examples in this quick guide and the normal way of using TCGA-Assembler to acquire data of all samples for a specific cancer type and assay platform.

Acquire miRNA Expression Data

The following command is included in QuickStartGuide_Examples.r to download miRNA expression data of six rectum adenocarcinoma samples measured by miRNA-seq assay.

```
miRNASeqRawData = DownloadmiRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-
2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform =
"miRNASeq", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-
6812-01", "TCGA-AF-2689-11", "TCGA-AF-2691-11"));
```

The first input argument gives the path of the directory traverse result file that will be used. The second input argument tells the function to recursively create folder QuickStartGuide_Results and subfolder RawData in the PWD, and the acquired data will be saved in the subfolder. The third input argument specifies acquiring data of rectum adenocarcinoma (READ), which can be replaced by other cancer types in TCGA including ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, SARC, SKCM, STAD, THCA, UCEC, UCS (see Supplementary Table 1). The fourth input argument indicates acquiring miRNA-seq data. The fifth input argument includes the TCGA barcodes of the six samples, for which we want to acquire data. Each barcode uniquely identifies a TCGA patient sample of a particular cancer type and sample type. For more information about TCGA sample barcodes, please refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode.

After the command is executed, in the RawData subfolder, two tab-delimited .txt files are generated,

READ__bcgsc.ca__illuminahiseq_mirnaseq__NCBI36__Jan-30-2014.txt
READ__bcgsc.ca__illuminahiseq_mirnaseq__GRCh37__Jan-30-2014.txt

They are both miRNA expression data of the six samples generated by the Illumina HiSeq 2000 Sequencing platform, but aligned to human reference genome Hg18 (indicated by NCBI36 in file name) and Hg19 (indicated by GRCh37 in file name), respectively. bcgsc.ca in the file name indicates Canada's Michael Smith Genome Sciences Centre, the institution that produced the data. illuminahiseq_mirnaseq indicates the assay platform. Jan-30-2014 indicates that the data are acquired using the directory traverse result obtained on Jan. 30th, 2014. In the files, the first column is miRNA names. Starting from the second column, two columns correspond to one sample, with one column being read count and the other column being Reads Per Million miRNA mapped (RPM), which is normalized expression value. The top row includes the full TCGA barcodes of samples.

If no TCGA sample barcodes are specified, i.e. the fifth input argument is omitted, in the DownloadmiRNASeqData function, miRNA-seq data of all READ patient samples will be acquired by default. The command should read as

```
miRNASeqRawData = DownloadmiRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform = "miRNASeq");
```

Providing no sample barcodes also works for other data acquisition functions to download data of all the samples belonging to the specified cancer category.

Acquire DNA Copy Number Data

In QuickStartGuide_Examples.r, the DownloadCNAData function is used to download DNA copy number data of six rectum adenocarcinoma (READ) samples. The command reads as

```
CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform = "genome_wide_snp_6", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2692-10", "TCGA-AG-4021-10"));
```

The second input argument of the function makes it recursively create folder QuickStartGuide_Results and folder RawData in the PWD, in which the acquired data will be saved. The fourth input argument genome_wide_snp_6 indicates the assay platform that was used to generate the data, which is Affymetrix® Genome-Wide Human SNP Array 6.0 that provides the most abundant DNA copy number data in TCGA. The fifth input argument gives the TCGA barcodes of the six samples, for which we want to acquire data. If no TCGA sample barcodes are specified, all level-3 DNA copy number data of READ samples will be acquired by default.

After the command is executed, we see four tab-delimited .txt data files in QuickStartGuide_Results/RawData, including

READ__broad.mit.edu__genome_wide_snp_6__hg18__Jan-30-2014.txt
READ__broad.mit.edu__genome_wide_snp_6__hg19__Jan-30-2014.txt
READ__broad.mit.edu__genome_wide_snp_6__nocnv_hg18__Jan-30-2014.txt
READ__broad.mit.edu__genome_wide_snp_6__nocnv_hg19__Jan-30-2014.txt

In the file names, broad.mit.edu indicates the Broad institute that generated the data. hg18 and hg19 indicate that in preparation for segmentation, the probes are sorted based on the order of reference genome Hg18 and

Hg19, respectively. nocnv indicates that a fixed set of probes that frequently contain germline CNVs are removed prior to segmentation. For details of the data generation pipeline of DNA copy number, please refer to TCGA description at https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/broad.mit.edu/genome_wide_snp_6/snp/broad.mit.edu_READ.Genome_Wide_SNP_6.mage-tab.1.2003.0/DESCRIPTION.txt. All four data files have the same format. Each row corresponds to a segment. Column one is full TCGA barcodes of samples. Column two is chromosome ID. Column three and four are the start and end positions of the segment, respectively. Column five is the number of probes in the segment. Column six is the copy number value.

Acquire mRNA Expression Data

In QuickStartGuide_Examples.r, the DownloadRNASeqData function is used to acquire normalized gene expression data and exon expression data of six READ samples generated by RNA-seq assay. The command reads as

```
RNASeqRawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform = "RNASeqV2", dataType = c("rsem.genes.normalized_results", "exon_quantification"), inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3732-11", "TCGA-AG-3742-11"));
```

The second input argument of the function tells it to create folder to save the acquired data. The fourth input argument indicates acquiring data generated by RNASeqV2 pipeline, which is one of the two post-processing pipelines that TCGA uses to process RNA sequencing reads. The fifth input argument indicates which types of data to acquire. rsem.genes.normalized_results refers to normalized gene expression values produced by the RSEM algorithm and exon_quantification refers to exon expression values produced by the RSEM algorithm. Refer to https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/unc.edu/illuminahiseq_rnaseqv2/rnaseqv2/unc.edu_READ.IlluminaHiSeq_RNASeqV2.mage-tab.1.6.0/DESCRIPTION.txt for details of RNASeqV2 pipeline and the algorithms.

After the data acquisition process finishes, two tab-delimited .txt data files are generated in QuickStartGuide_Results/RawData, including

```
READ__unc.edu__illuminahiseq_rnaseqv2__exon_quantification__Jan-30-2014.txt
READ__unc.edu__illuminahiseq_rnaseqv2__rsem.genes.normalized_results__Jan-30-2014.txt
```

In the file names, unc.edu indicates the University of North Carolina that generated the data. illuminahiseq_rnaseqv2 indicates the Illumina Hiseq 2000 sequencing platform and RNASeqV2 post-processing pipeline. In the exon_quantification data file, the first column gives the exon genomic coordinate. Starting from the second column, three columns correspond to one sample, which give raw base counts, coverage, and RPKM values of exon expression. Refer to the above link for details of the calculations. In the rsem.genes.normalized_results data file, the first column gives gene symbol and Entrez ID (separated by "|") and the other columns are gene expression values of normalized counts produced by the RSEM algorithm.

Acquire DNA Methylation Data

TCGA uses two assays to measure DNA methylation, including Illumina HumanMethylation27 BeadChip and Illumina HumanMethylation450 BeadChip. The DownloadMethylationData function can acquire both HumanMethylation27 data and HumanMethylation450 data. In QuickStartGuide_Examples.r, HumanMethylation27 data of six READ samples are acquired using the following command.

Methylation27RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData/", cancerType = "READ", assayPlatform = "humanmethylation27", inputPatientIDs = c("TCGA-AG-3583-01", "TCGA-AG-A032-01", "TCGA-AF-2692-11", "TCGA-AG-4001-01", "TCGA-AG-3608-01", "TCGA-AG-3574-01"));

After the data acquisition completes, a tab-delimited .txt data file READ__jhu-usc.edu__humanmethylation27__Jan-30-2014.txt is generated in QuickStartGuide_Results/RawData. jhu-usc.edu in the file name indicates Johns Hopkins University and The University of Southern California that produced the data. In the file, the first column is the Illumina ID of CpG site; the second column is gene symbol; the third column is chromosome ID; the fourth column is genomic coordinate. The other columns are samples with full TCGA barcodes shown in the top row. In QuickStartGuide_Examples.r, HumanMethylation450 data are acquired using the following command.

Methylation450RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData", cancerType = "READ", assayPlatform = "humanmethylation450", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-A01W-11", "TCGA-AG-3731-11"));

After the command is executed, another tab-delimited .txt data file READ__jhu-usc.edu__humanmethylation450__Jan-30-2014.txt is generated, which has the same file format as the HumanMethylation27 data file described above.

Acquire Protein Expression Data

In QuickStartGuide_Examples.r, the following command uses the DownloadRPPAData function to acquire Reverse Phase Protein Array (RPPA) protein expression data of six READ patient samples

RPPARawData = DownloadRPPAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData", cancerType = "READ", assayPlatform = "mda_rppa_core", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3582-01", "TCGA-AG-4001-01"));

A tab-delimited .txt file is created in QuickStartGuide_Results/RawData. The file name is READ__mdanderson.org__mda_rppa_core__Jan-30-2014.txt, where mdanderson.org indicates MD Anderson Cancer Center that generated the data and mda_rppa_core indicates the RPPA assay used to produce the data. In the file, the top row shows the full TCGA sample barcodes. The first column includes protein antibody name (after "|") and corresponding gene symbols (before "|"). The other columns are normalized protein expression data of samples. For details about how the RPPA data were generated and normalized, please refer to TCGA description at https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/mdanderson.org/mda_rppa_core/protein_exp/mdanderson.org_READ.MDA_RPPA_Core.Level_3.1.0.0/DESCRIPTION.txt

Acquire De-identified Patient Clinical Information

In QuickStartGuide_Examples.r, the following command uses the DownloadClinicalData function to acquire clinical information of READ patient samples.

DownloadClinicalData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./QuickStartGuide_Results/RawData", cancerType = "READ", clinicalDataType = c("patient", "drug", "follow_up"));

The fourth input argument of the DownloadClinicalData function controls which types of clinical information should be acquired. patient indicates patient information including survival, cancer grades, and others. drug indicates patient drug treatment information. follow_up indicates patient follow up information. After the command is executed, there are three files in QuickStartGuide_Results/RawData, including

clinical_drug_read.txt
clinical_patient_read.txt
clinical_follow_up_v1.0_read.txt

For details of the formats of clinical information, please check TCGA website.

5.2 Introduction of Basic Data Processing Functions in Module B.

All level-3 data files acquired by Module A need to be processed by the basic processing functions of Module B. These basic processing functions are ProcessCNAData, ProcessmiRNASeqData, ProcessMethylation27Data, ProcessMethylation450Data, ProcessRNASeqData, and ProcessRPPADataWithGeneAnnotation, all of which have a name beginning with "Process". These functions do basic processing on the data acquired from TCGA DCC, including extraction of most useful measurements, data quality check, and others. For example, gene symbols in TCGA data are checked and corrected. Gene symbols that are errors caused by auto-conversion of Excel when the raw data were produced by various labs, such as FEB6 being transferred to 6-FEB, and alias/obsolete gene symbols are mapped to official HGNC gene symbols. Boxplots are drawn for the purpose of identifying and removing sample outliers. If a genomic feature corresponds to multiple genes, its measurement values (a row in the data matrix) is duplicated for each gene it associates with. For DNA copy number data, gene-level copy number value, which is the average copy number of the genomic region of a gene, is calculated. The basic data processing functions are also compatible with Firehose data files meaning that they can process level-3 TCGA data files downloaded from Firehose website, do the data quality check and transform mentioned above, and import them into R for subsequent analysis. Refer to TCGA-Assembler User Manual for details of using Module B functions to process Firehose data files.

In TCGA-Assembler Module B, a dataset is usually represented by two matrix variables called Des and Data or a list object formed by these two variables. Des is a character matrix containing the descriptions of genomic features and Data is a numeric matrix including the measurement values. Des and Data have the same number of rows. Each row of Des includes the description of a genomic feature whose measurements are in the corresponding row of Data. Each column in Data is a sample with its TCGA sample barcode as the column name.

Process miRNA Expression Data

The command of using ProcessmiRNASeqData in QuickStartGuide_Examples.r is the following.

```
miRNASeqData = ProcessmiRNASeqData(inputFilePath =
"./QuickStartGuide_Results/RawData/READ__bcgsc.ca__illuminahiseq_mirnaseq__GRCh37__Jan-30-2014.txt",
outputFileName = "READ__illuminahiseq_mirnaseq", outputFileFolder =
"./QuickStartGuide_Results/BasicProcessingResult");
```

The first input argument inputFilePath is the path of the miRNA expression data file that is acquired by Module A and needs processing. outputFileName is a string to form the names of processed data files. outputFileFolder specifies a folder to save the processed data files. After the command is executed, a subfolder named BasicProcessingResult is created in the QuickStartGuide_Results folder with four files in it.

READ__illuminahiseq_mirnaseq__ReadCount.rda

READ__illuminahiseq_mirnaseq__ReadCount.txt

contain the same miRNA read count dataset, but in two different file formats for the convenience of using the data in different software environments. In READ__illuminahiseq_mirnaseq__ReadCount.rda file, the dataset is represented by two matrix variables Des and Data, as introduced above. The other two files are

READ__illuminahiseq_mirnaseq__RPM.rda
READ__illuminahiseq_mirnaseq__RPM.txt

which contain the RPM values of miRNAs. Also, in READ__illuminahiseq_mirnaseq__RPM.rda, the miRNA expression dataset is represented by two matrix variables Des and Data. The function returns a list object of the RPM values to miRNASeqData, formed by Des and Data.

Process DNA Copy Number Data

In QuickStartGuide_Examples.r, the following command does basic processing on level-3 DNA copy number data acquired by Module A.

```
READ.GeneLevel.CNA = ProcessCNAData(inputFilePath =
"./QuickStartGuide_Results/RawData/READ__broad.mit.edu__genome_wide_snp_6__hg19__Jan-30-2014.txt",
outputFileName = "READ__genome_wide_snp_6__GeneLevelCNA", outputFileFolder =
"./QuickStartGuide_Results/BasicProcessingResult", refGenomeFile =
"./SupportingFiles/Hg19GenePosition.txt");
```

The input argument inputFilePath is the path of the data file acquired by Module A and to be processed. outputFileName is a string to form the names of the processed data files. outputFileFolder specifies a folder to save the processed data files. refGenomeFile specifies the supporting file containing gene genomic positions to be used by the function. Because the input copy number data were generated based on reference genome Hg19, the supporting file to be used should be Hg19GenePosition.txt in the SupportingFiles folder included in the package. The ProcessCNAData function calculates gene-level copy number values, which are the average copy number of the genomic region of a gene. After the command is executed, in the BasicProcessingResult subfolder, three files are generated, including

READ__genome_wide_snp_6__GeneLevelCNA.rda
READ__genome_wide_snp_6__GeneLevelCNA.txt
READ__genome_wide_snp_6__GeneLevelCNA__boxplot.png

The .rda and .txt files include the same gene-level copy number data but in different file formats. The .png file is a boxplot picture of the gene-level copy number data. In the .rda file, the gene-level copy number data are represented by two matrix variables Des and Data. Des gives gene descriptions (including gene symbol, chromosome ID, and strand) and Data contains the copy number values. The function returns a list object formed by Des and Data to READ.GeneLevel.CNA.

Process mRNA Expression Data

In QuickStartGuide_Examples.r, the following command does basic processing of normalized gene expression data acquired by Module A.

```
GeneExpData = ProcessRNASeqData(inputFilePath =
"./QuickStartGuide_Results/RawData/READ__unc.edu__illuminahiseq_rnaseqv2__rsem.genes.normalized_res
```

ults__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_rnaseqv2__GeneExp", outputFileFolder = "./QuickStartGuide_Results/BasicProcessingResult", dataType = "GeneExp", verType = "RNASeqV2");

The input argument dataType specifies the type of data to be processed. Available options include "GeneExp" (gene expressions) and "ExonExp" (exon expressions). verType specifies which post-processing pipeline was used to generate the data. This command generates three files in the BasicProcessingResult subfolder, including

READ__illuminahiseq_rnaseqv2__GeneExp.rda
READ__illuminahiseq_rnaseqv2__GeneExp.txt
READ__illuminahiseq_rnaseqv2__GeneExp__boxplot.png

The .rda and .txt files contain the same gene expression data in different file formats. The .png file is a boxplot picture of the gene expression dataset for the purpose of identifying and removing outlier samples. In the .rda file, Des contains gene descriptions (including gene symbol and Entrez ID) and Data contains the normalized read counts of mRNAs. The function returns a list object to GeneExpData, formed by Des and Data.

The same function is also used to process exon expression data acquired by Module A. The command reads as

ExonExpData = ProcessRNASeqData(inputFilePath = "./QuickStartGuide_Results/RawData/READ__unc.edu__illuminahiseq_rnaseqv2__exon_quantification__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_rnaseqv2__ExonExp", outputFileFolder = "./QuickStartGuide_Results/BasicProcessingResult", dataType = "ExonExp", verType = "RNASeqV2");

The ProcessRNASeqData function extracts RPKM values of exon expressions from the input data file and saves the extracted data in the BasicProcessingResult subfolder. The data files are

READ__illuminahiseq_rnaseqv2__ExonExp.rda
READ__illuminahiseq_rnaseqv2__ExonExp.txt

The exon RPKM values are also returned by the function in a list object to ExonExpData, which is composed of Des and Data. Des contains genomic locations of exons and Data contains the RPKM expression values of exons with TCGA sample barcodes as its column names.

Process DNA Methylation Data

In QuickStartGuide_Examples.r, the ProcessMethylation27Data function and the ProcessMethylation450Data function are used to process the HumanMethylation27 data and HumanMethylation450 data acquired by Module A, respectively. If in the methylation data a CpG site corresponds to more than one gene, the measurements of the CpG site (a row in the data matrix) will be duplicated for each gene associated with the CpG site. The command of processing HumanMethylation27 data is as following.

Methylation27Data = ProcessMethylation27Data(inputFilePath = "./QuickStartGuide_Results/RawData/READ__jhu-usc.edu__humanmethylation27__Jan-30-2014.txt", outputFileName = "READ__humanmethylation27", outputFileFolder = "./QuickStartGuide_Results/BasicProcessingResult");

This command generates three files in the BasicProcessingResult subfolder, including

READ__humanmethylation27.rda
READ__humanmethylation27.txt

READ__humanmethylation27__boxplot.png

which are the processed dataset in two different file formats and its boxplot picture. The processed data are methylation values of CpG sites and the description of CpG sites include Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. The command of processing HumanMethylation450 data reads as

```
Methylation450Data = ProcessMethylation450Data(inputFilePath =
"./QuickStartGuide_Results/RawData/READ__jhu-usc.edu__humanmethylation450__Jan-30-2014.txt",
outputFileName = "READ__humanmethylation450", outputFileFolder =
"./QuickStartGuide_Results/BasicProcessingResult");
```

This command also generates three files in the BasicProcessingResult subfolder, including

```
READ__humanmethylation450.rda
READ__humanmethylation450.txt
READ__humanmethylation450__boxplot.png
```

Process Protein Expression Data

The ProcessRPPADataWithGeneAnnotation function is used to process the RPPA protein expression data acquired by Module A. If a protein is encoded by more than one gene, this function will duplicate the measurements of the protein (a row in the data matrix) for each gene that encodes the protein. The command reads as

```
RPPAData = ProcessRPPADataWithGeneAnnotation(inputFilePath =
"./QuickStartGuide_Results/RawData/READ__mdanderson.org__mda_rppa_core__Jan-30-2014.txt",
outputFileName = "READ__mda_rppa_core", outputFileFolder =
"./QuickStartGuide_Results/BasicProcessingResult");
```

It generates three files in the BasicProcessingResult subfolder, including

```
READ__mda_rppa_core.rda
READ__mda_rppa_core.txt
READ__mda_rppa_core__boxplot.png
```

The .rda and .txt files include the protein expression data. The .png file is a boxplot picture of the data. In the .rda file, Des contains the description of proteins (including genes encoding the protein and the protein antibody name) and Data contains protein expression values. A list object of Des and Data is returned by the function to RPPAData.

5.3 Introduction of Advanced Data Processing Functions in Module B.

Data outputted from the basic data processing functions can be further processed by advanced data processing functions in Module B to fulfill various data manipulation needs. The advanced data processing functions include ExtractTissueSpecificSamples for extracting data of the samples belonging to user specified tissue types, MergeMethylationData for merging two DNA methylation datasets generated by either the same or different Illumina HumanMethylation BeadChips, CalculateSingleValueMethylationData for calculating an average methylation value of CpG sites in a particular region of gene, and CombineMultiPlatformData that combines multi-platform data for integrative data analysis.

Combine Multi-platform Data

CombineMultiPlatformData combines multi-platform TCGA data for integrative analysis, i.e. generating a single mega data table by combining multiple files that include data generated by different genomic and epigenomic features measured by different assays. It has two different combination approaches. One is to identify samples measured by all assay platforms and merge the multi-platform data of these common samples, which is actually the default setting of the function. The other is to include a sample as long as it is measured by at least one assay platform. The merged data are then sorted by genes such that data of the multiple features of a gene are stacked next to each other in the data table. In QuickStartGuide_Examples.r, CombineMultiPlatformData is used to combine the miRNA expression data, gene expression data, protein expression data, gene-level copy number data, and single-value methylation data of several rectum adenocarcinoma (READ) samples. The command reads as

```
MergedData = CombineMultiPlatformData(inputDataList = inputDataList);
```

The input argument inputDataList is a vector of list objects. Each element in the vector is a list object of three variables Des, Data, and dataType, which represent one dataset to be combined. Des is a character matrix including descriptions of genomic features and Data is a numeric matrix including the data, as introduced previously. dataType is a string indicating the type of data. Options of dataType include "GeneExp", "ProteinExp", "Methylation", "CNA", and "miRNAExp", standing for gene expression, protein expression, DNA methylation, DNA copy number, and miRNA expression, respectively. The combined data are saved as a tab-delimited .txt file named as CombinedMultiPlatformData.txt in the AdvancedProcessingResult subfolder.

	A	B	C	D	E	F	G	H
1	GeneSymbol	Platform	Description	TCGA-DC-5869-01	TCGA-EI-6884-01	TCGA-F5-6812-01	TCGA-G5-6572-01	
2	A1BG	CN	CHR19-	-0.1539	-0.0097	0.0324	-0.0275	
3	A1BG	GE	1	22.7014	23.7136	24.1244	29.3094	
4	A1BG	ME	All|Both	0.451386172	0.593891737	0.472737373	0.647319654	
5	A1BG-AS1	CN	CHR19+	-0.1539	-0.0097	0.0324	-0.0275	
6	A1BG-AS1	GE	503538	7.8534	31.0872	22.4338	16.0097	
7	A1BG-AS1	ME	All|Both	0.564498105	0.743445466	0.616766894	0.698793372	
8	A1CF	CN	CHR10-	-0.0481	-0.0231	0.015	-0.043	
9	A1CF	GE	29974	201.1577	140.0447	67.9093	299.6769	
10	A1CF	ME	All|Both	0.546876073	0.647204087	0.673238513	0.585206663	
11	A2M	CN	CHR12-	-0.086	-0.0038	-0.1623	-0.0499	
12	A2M	GE	2	3941.1095	13495.302	17012.8458	6446.2197	
13	A2M	ME	All|Both	0.480743452	0.754819342	0.757780361	0.703485429	
14	A2M-AS1	CN	CHR12+	-0.086	-0.0038	-0.1623	-0.0499	
15	A2ML1	CN	CHR12+	-0.086	-0.0038	-0.1623	-0.0499	
16	A2ML1	GE	144568	0	0	0	0	
17	A2ML1	ME	All|Both	0.524795863	0.705412904	0.734093963	0.707895478	
18	A2MP1	CN	CHR12-	-0.086	-0.0038	-0.1623	-0.0499	
19	A4GALT	CN	CHR22-	-0.1187	-0.0117	-0.4066	-0.0421	

We can open CombinedMultiPlatformData.txt using Excel, and look at the combined data table as shown above. The first column of the table shows gene symbols. The second column shows the platforms, which include copy number (CN), gene expression (GE), methylation (ME), protein expression (PE), and miRNA expression (miRExp), although PE and miRExp are not seen in the top rows of the table. The third column gives additional description of the features. For CN platform it is chromosome ID and strand. For GE platform it is gene Entrez ID. For ME platform it indicates how the single methylation value is calculated (see ***Calculate Single-Value Methylation Data*** for details). For PE platform, it is the name of the protein antibody used in the assay. The

other four columns are data of the four samples that are measured by all five different assay platforms. The multi-platform data of a gene are adjacent rows in the table.

Extract Tissue Type Specific Data

TCGA data usually include samples of multiple tissue types, such as primary solid tumor, recurrent solid tumor, and blood derived normal cells. For a full list of TCGA tissue types, please look at https://tcga-data.nci.nih.gov/datareports/codeTablesReport.htm?codeTable=Sample%20type. Data analyzers may be interested in studying the data of particular tissue types. The ExtractTissueSpecificSamples function allows users to conveniently extract the data of samples belonging to specified tissue types. In QuickStartGuide_Examples.r, the following command extracts data of primary solid tumors from the HumanMethylaton450 data of rectum adenocarcinoma (READ) samples.

```
ExtractedData_TP = ExtractTissueSpecificSamples(inputData = Methylation450Data$Data, tissueType = "TP", singleSampleFlag = FALSE , sampleTypeFile = "./SupportingFiles/TCGASampleType.txt");
```

The input argument inputData is a data matrix from which tissue type specific data will be extracted. tissueType is a string or string vector indicating the tissue types of interest. "TP" indicates primary solid tumors. For indicators of all different tissue types, please see the introduction of ExtractTissueSpecificSamples function in TCGA-Assembler User Manual. sampleTypeFile indicates the path of the TCGA sample type file to be used by the function. It is TCGASampleType.txt in the SupportingFiles folder in the package. After the command is executed, data of four primary solid tumors are extracted, each having methylation values of 526742 CpG sites over the genome. The following command extracts data of both primary solid tumors ("TP") and solid normal tissues indicated by "NT".

```
ExtractedData_TP_NT = ExtractTissueSpecificSamples(inputData = Methylation450Data$Data, tissueType = c("TP", "NT"), singleSampleFlag = TRUE , sampleTypeFile = "./SupportingFiles/TCGASampleType.txt");
```

Six samples including four primary solid tumors and two solid normal tissues are extracted.

Calculate Single-Value Methylation Data

TCGA methylation data usually have measurements at multiple CpG sites of a gene. Especially, HumanMethylation450 BeadChip usually measures 10~30 CpG sites in different regions of a gene. In many studies, we may need a single value to summarize the methylation level of a particular region of genes. The CalculateSingleValueMethylationData function is designed for this purpose. It calculates the average methylation values of CpG sites in a specified region of each gene, such as within 200 base pairs of Transcription Start Site (TSS200) and 3' untranslated region (3'UTR). The following command in QuickStartGuide_Examples.r calculates an average methylation value of CpG sites that are within 1500 base pairs of TSS and are DNAse hypersensitive.

```
Methylation450_TSS1500_DHS = CalculateSingleValueMethylationData(input = Methylation450Data, regionOption = "TSS1500", DHSOption = "DHS", outputFileName = "READ__humanmethylation450__SingleValue", outputFileFolder = "./QuickStartGuide_Results/AdvancedProcessingResult", chipAnnotationFile = "./SupportingFiles/MethylationChipAnnotation.rda");
```

The input argument input is a list object formed by Des and Data that represents the methylation dataset for which single-value data need to be calculated. regionOption = "TSS1500" indicates that only CpG sites within 1500 base pairs of TSS are included in calculation. DHSOption = "DHS" indicates that only CpG sites hypersensitive to DNAse are included. chipAnnotationFile indicates the path of the chip annotation file to be

used by the function, which is the MethylationChipAnnotation.rda file in the SupportingFiles folder in the package. Three files are generated by this command in the AdvancedProcessingResult subfolder, including

READ__humanmethylation450__SingleValue__TSS1500__DHS.rda
READ__humanmethylation450__SingleValue__TSS1500__DHS.txt
READ__humanmethylation450__SingleValue__TSS1500__DHS__boxplot.png

The .rda file and .txt file contain the single-value methylation data. The .png file is a boxplot picture of the single-value data. In the .rda file, Des includes gene symbol, and the regionOption and DHSOption used to calculate the data. Data includes the calculated single-value methylation data. A list object formed by Des and Data is returned to Methylation450_TSS1500_DHS.

Another command in QuickStartGuide_Examples.r calculates an average methylation value of all CpG sites regardless of their genomic region and sensitivity to DNAse for each gene, using regionOption = "All" (indicating all regions) and DHSOption = "Both" (indicating both DNAse hypersensitive and not). The command is

```
Methylation450_OverallAverage = CalculateSingleValueMethylationData(input = Methylation450Data, regionOption = "All", DHSOption = "Both", outputFileName = "READ__humanmethylation450__SingleValue", outputFileFolder = "./QuickStartGuide_Results/AdvancedProcessingResult", chipAnnotationFile = "./SupportingFiles/MethylationChipAnnotation.rda");
```

This command also generates three files in the AdvancedProcessingResult subfolder, including

READ__humanmethylation450__SingleValue__All__Both.rda
READ__humanmethylation450__SingleValue__All__Both.txt
READ__humanmethylation450__SingleValue__All__Both__boxplot.png

Combine Methylation Datasets

HumanMethylation27 BeadChip measures methylation at ~27,000 CpG sites and HumanMethylation450 BeadChip measures methylation at ~450,000 CpG sites. More than 90% of the CpG sites measured by HumanMethylation27 BeadChip are also measured by HumanMethylation450 BeadChip. For many cancer types, such as READ, both HumanMethylation27 BeadChip and HumanMethylation450 BeadChip have been used to generate data for a large number of samples. Data from both groups may be included in downstream statistical analysis for increased power. The MergeMethylationData function can combine two methylation datasets into one. The function first identifies the CpG sites included in both datasets and combines the data of these common CpG sites. Quantile normalization is then performed on the combined data to eliminate any systematic difference (if any) between the two datasets. In QuickStartGuide_Examples.r, the following command combines the HumanMethylation27 data and HumanMethylation450 data of several READ samples prepared by the basic data processing functions.

```
Methylation27_450_Merged = MergeMethylationData(input1 = Methylation27Data, input2 = Methylation450Data, outputFileName = "READ__humanmethylation27_450_merged", outputFileFolder = "./QuickStartGuide_Results/AdvancedProcessingResult");
```

The input arguments input1 and input2 are two list objects of Des and Data, representing the two methylation datasets to be combined. Four files are generated by this command in the AdvancedProcessingResult subfolder, including

READ__humanmethylation27_450_merged.rda

READ__humanmethylation27_450_merged.txt
READ__humanmethylation27_450_merged__BeforeNormalizationBoxplot.png
READ__humanmethylation27_450_merged__AfterNormalizationBoxplot.png

The .rda file and .txt file contain the combined and normalized data. The .png files are boxplots of combined data before and after normalization.